A COMPARISON OF DISCRETE INVERSE METHODS FOR DETERMINING PARAMETERS OF AN ECONOMIC MODEL.

By

Caleb Jurkowski, A.B.

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in

Mathematics

University of Alaska Fairbanks

August 2017

APPROVED:

David Maxwell, Committee Chair
Margaret Short, Committee Member
Edward Bueler, Committee Member
Leah Wrenn Berman, Chair
Department of Mathematics and Statistics

Abstract

We consider a time-dependent spatial economic model for capital in which the region's production function is a parameter. This forward model predicts the distribution of capital of a region based on that region's production function. We will solve the inverse problem based on this model, i.e. given data describing the capital of a region we wish to determine the production function through discretization. Inverse problems are generally ill-posed, which in this case means that if the data describing the capital are changed slightly, the solution of the inverse problem could change dramatically. The solution we seek is therefore a probability distribution of parameters. However, this probability distribution is complex, and at best we can describe some of its features.

We describe the solution to this inverse problem using two different techniques, Markov chain Monte Carlo (Metropolis Algorithm) and least squares optimization, and compare summary statistics coming from each method.

Table of Contents

I would like to thank my advisor Dr. David Maxwell for his abundant time, patience, and guidance. I would also like to thank my committee members Dr. Margaret Short and Dr. Ed Bueler for their time and energy in carefully reading drafts and giving feedback.

Chapter 1

Introduction and Background

1.1 Mathematical modeling

Consider a mathematical model that predicts a measurable quantity $\mathbf{d}$. The model itself depends on some physical law or process, while the outcomes $\mathbf{d} \in \mathbb{R}^\ell$ are dependent on the parameters $\mathbf{m} \in \mathbb{R}^k$ of that model by a function $\mathbf{g}(\mathbf{m}) = \mathbf{d}$. We will call computing $\mathbf{g}(\mathbf{m})$ the *forward problem* or *model* (from here we refer to the forward problem as $\mathbf{g}(\mathbf{m})$ or simply $\mathbf{g}$). The *inverse problem* is naturally the opposite of the forward problem: given actual measurements $\mathbf{d}$, we wish to describe the parameters of the model.

The forward problem is usually well-posed, i.e., it has a unique solution that depends continuously on the parameters of the model. However, the inverse problem is often ill-posed. The model parameters do not depend continuously on the data; small errors or perturbations of the data $\mathbf{d}$ can lead to large changes in the model parameters $\mathbf{m}$. Given two similar collections of data, the model parameters found by solving the inverse problem may be quite different. Indeed, an inverse problem may not have a unique solution. To compensate for these difficulties, the solution to an inverse problem can generally be thought of as a probability distribution.

However, as is often the case with solving inverse problems, one final "answer" is desirable (the one set of parameters that *best* fit the data). One approach to solving inverse problems by obtaining a single set of parameters is regularization. A standard method is Tikhonov regularization. This method combines minimizing the distance between the data and the forward problem solution by adding a regularization or "smoothing" term to deal with the ill-posedness. But this single solution does not come with a description of how likely it is, so a probability distribution may be a better way to describe the solution.

Indeed, the approach that will be taken in this paper is of the probabilistic variety. The solution we seek is a probability distribution P (accompanied by probability density function φ). The probability that the model parameters lie in a region A is given by

$$P(A) = \int_A \varphi(\mathbf{m}) d\mathbf{m}. \tag{1.1}$$

However, just knowing the probability density function is not enough; if $\mathbf{m}$ is of high dimension, we can not visualize this solution. Therefore, we want to find a way to describe features of the probability distribution, such as the mean, mode, covariances, etc.

Mathematical modeling has long been a mainstay of social science fields such as economics. Increasingly, inverse problem theory is being used to solve problems in economics [4]. We will consider such a model and inverse problem, and we will use two different methods (one inherently probabilistic, the other in the regularization vein) to estimate properties of the solution.

1.2 Spatial model of capital assets

In 1956, George Solow [2] developed a model for capital k (GDP) of a region/state whose parameter was that region's production function. This time-dependent model, subject to initial conditions, is

$$\frac{d}{dt}k(t) = a(t)f(k(t)) - \delta k(t) \tag{1.2}$$

where $a(t)$ is the technology level, $f : \mathbb{R}^+ \to \mathbb{R}^+$ is the production function (dependent on capital), and $\delta \in \mathbb{R}$ is depreciation. Capasso et al. [2] used this to develop a spatial model where capital depends on space as well as time. Consider $\mathbf{x} \in \Omega \subseteq \mathbb{R}^n$ for $n \in \{1, 2\}$ as the location of households, and time $t \in [0, T]$. In addition to initial conditions, we will now need boundary conditions. This time-dependent partial differential equation model is

$$\begin{cases} \dfrac{\partial}{\partial t}k(\mathbf{x}, t) = \Delta k(\mathbf{x}, t) + a(\mathbf{x}, t)f(k(\mathbf{x}, t)) - \delta k(\mathbf{x}, t) \\ \nabla k \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega \times [0, T], \end{cases} \tag{1.3}$$

where Δ is the Laplacian ($\partial_{x_1}^2 k + \partial_{x_2}^2 k$ in two dimensions and just $\partial_x^2 k$ in one dimension) and the Neumann boundary condition $\nabla k \cdot \mathbf{n} = 0$ models no flow of capital across the boundary (i.e., a closed economy). Note the diffusion coefficient is 1 which depends on the time scaling.

If we restrict ourselves to one spatial dimension ($x \in [0, L] \subseteq \mathbb{R}$), (1.3) becomes

$$\begin{cases} \dfrac{\partial}{\partial t}k(x,t) = \dfrac{\partial^2}{\partial x^2}k(x,t) + a(x,t)f(k(x,t)) - \delta k(x,t) \\ k_x(0,t) = k_x(L,t) = 0. \end{cases} \tag{1.4}$$

Assuming a and δ are known, this forward model predicts the value of capital k at a given point x at time t, The parameter of interest is f, and so we define the forward problem

$$\mathbf{g}(f) = k, \tag{1.5}$$

whose solution for any given f is the solution of the nonlinear partial differential equation (1.4). Of course before we proceed, it is important to note that this problem is well-posed.

Theorem 1. *Let $k(x,0) \in C^2[0,L]$, $\|k(x,0)\| < \infty$, and $f \in C^1(\mathbb{R}^+)$ be Lipschitz continuous. Furthermore, let $a \in C([0,L] \times [0,T])$. Equation (1.4) has a unique classical solution k.*

Proof. This is Theorem 2.1 of [3]. □

The solution of this equation is interesting if it is the capital we wish to predict at a later time and place. However it is the identification of the parameter f that we are interested in obtaining. An accurate description of the production function is necessary in order for the model to make good predictions of GDP or growth.

1.3 The inverse problem

Let us now define the inverse problem at hand. Engbers et al. [3] solved the inverse problem (1.5) using so-called Tikhonov regularization, with an arbitrarily chosen regularization parameter. We will make the same assumptions as were made in [3] as we attempt to solve the same inverse problem via different means. A few such assumptions are the following: the model (1.4) is an exact mathematical representation of a (one-dimensional) region's capital, the domain for the function k describing capital is the same, a mathematical function f represents a region's production function, and f is convex for small values of k and concave for larger values of k. One last common assumption - - - to facilitate our goal of the recovery of the production function, we will take $a(x,t)$ and δ in (1.4) to be known. However, our interpretation of "solution" will be different. We wish to find and describe some features of a probability distribution of functions f instead of just a single function.

The production function of a firm (or nation/state/etc.) is a relationship between the quantity of inputs that entity uses and the quantity of output [5]. Some factors of production are land, labor, technology, and capital. Solow and others [2],[3] simplify the model by describing production as a mathematical function of capital and labor. As the authors of [2] and [3] tie labor with capital, we will do so also, and we are left with our current state of the production function, f, as a function solely of capital k.

The identification of the production function of a region (or state/nation) is vital to predict long term growth. In addition to this, knowing the production function can also help drive decisions of governments (or corporations) such as resource or asset allocation. In this regard, a spatial model for capital (or growth) could be advantageous.

The standard neoclassical representation of the production function f is nonnegative, increasing, and concave [3]. It also satisfies the Inada conditions [3]:

$$(1)\ f(0) = 0, \quad (2)\ \lim_{k\to\infty} f'(k) = 0, \quad (3)\ \lim_{k\to 0^+} f'(k) = \infty.$$

These conditions are satisfied by the class of Cobb-Douglas production functions $f(k) = k^\alpha$ where $\alpha \in (0, 1)$ [3], an example of which is shown in Figure 1.1. However just like Engbers et al., we shall disregard the third Inada condition and seek a convex-concave production function. Therefore we must define a set which contains all possible production functions that satisfy the first two conditions.

Let $K > 0$ and $f'_{\max} > 0$ be constant. For $I = [0, K]$, we define our set of admissible functions as

$$\mathcal{F} = \{f \in H^1[I] \,|\, f(0) = 0,\ 0 \le f'(k) \le f'_{\max}\} \tag{1.6}$$

where H^1 is the set of all functions in L^2 whose weak derivatives are also in L^2. An example of a family of convex-concave functions that belong to $\mathcal{F}$ is given by

$$f(k) = \frac{\alpha_1 k^\beta}{1 + \alpha_2 k^\beta}.$$

Figure 1.2 shows such a function when $\alpha_1 = \alpha_2 = 0.0005$ and $\beta = 4$.

Apart from the condition $0 \le f'(k) \le f'_{\max}$, $f(0) = 0$ will be important to our method of solving this inverse problem. This condition implies there can be no growth without some capital resources.

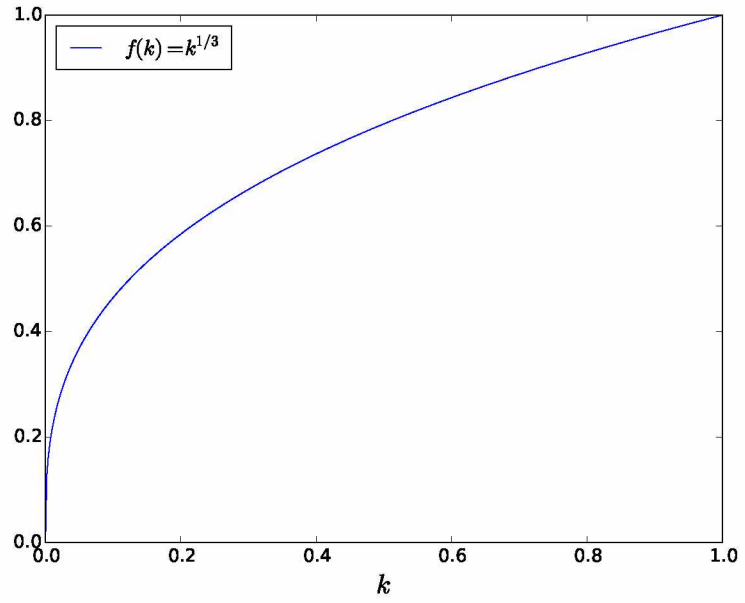

Figure 1.1: A Cobb-Douglas production function $f(k) = k^{1/3}$

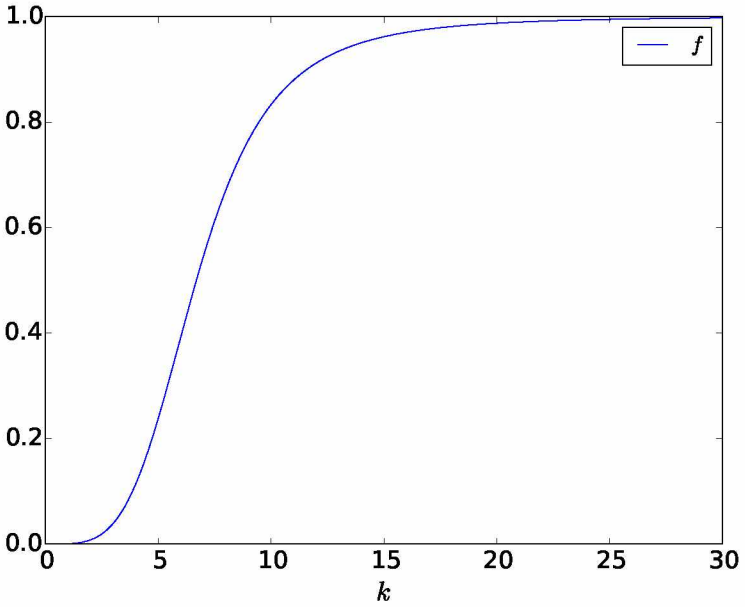

Figure 1.2: $f(x) = \dfrac{0.0005k^4}{1 + 0.0005k^4}$

1.4 Determining parameters of the model

We need to discretize the problem to numerically solve the inverse problem. Engbers et al. [3] minimize

$$J_\beta(f) = \|\mathbf{g}(f) - k\|_{L^2}^2 + \beta\|f - f_{\text{prior}}\|_{H^1(I)}^2,$$

where $\beta\|f - f_{\text{prior}}\|$ is a Tikhonov regularization term, to solve the inverse problem. They find a linear piecewise function for f in the end, but they solve the forward problem $\mathbf{g}(f)$ for a function f along the way. We will instead discretize the problem from the beginning (solve the forward problem given a vector instead of a function) as we seek a probability distribution as an answer in the end - - - not just a single function.

Consider some $n \in \mathbb{Z}^+$. We will discretize the forward problem by first approximating f by a linear piecewise function $\tilde{f}_n$ by evaluating f at n equally spaced nodes with spacing Δk. Note the first node is at $k = \Delta k$, not $k = 0$. We know $f(0) = 0$ from the conditions of $\mathcal{F}$, so we need not consider that point. The plot of an example f, along with $\tilde{f}_5$ for $n = 5$ nodes is shown in Figure 1.3.

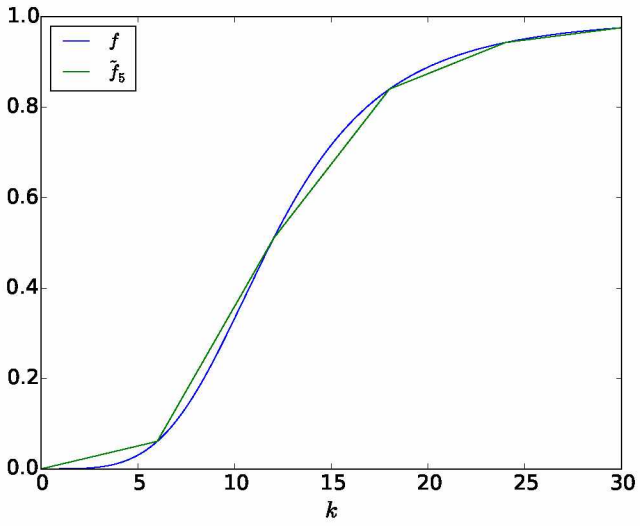

Figure 1.3: Plot of an $f \in \mathcal{F}$ and its approximate linear piecewise function $\tilde{f}_5$ for $n = 5$

Now we can finally discretize the forward problem: instead of solving $\mathbf{g}(f)$, we will solve $\mathbf{g}(\mathbf{m})$, where $\mathbf{m} \in M = [0, f'_{\max}]^n \subseteq \mathbb{R}^n$ is a vector whose components are the slopes between adjacent nodes of $\tilde{f}_n$. As $f(0) = 0$, there is a bijective map between $\mathbf{m}$ and $\tilde{f}$. It is this correspondence that will allow us to solve the forward problem (1.4) if we are given either $\mathbf{m}$ or $\tilde{f}$.

1.5 Probabilistic approach to inverse problems

As we alluded to earlier, the solution to an inverse problem in general is a probability distribution on the space of model parameters [9]. We will describe how to use the forward model (1.4) and given data to transform an a priori distribution on the model space into our solution, the a posteriori probability distribution. This transformation comes from a combination of the states of information describing our prior knowledge and data and from our understanding of the theory described by our model. In order to describe the posterior distribution, we must first define the necessary objects as well as the transformation of the a priori distribution.

Depending on the context of the problem, the model parameters are elements of the model space M. Similarly, the observed data measures belong to the data space D. Frequently M and D have the structure of a vector space, in which case we say they are *linear*. In the context of our particular problem, $M = [0, f'_{\text{max}}]^n$, as defined in the previous section, and $D = \mathbb{R}^p$, where $p \in \mathbb{Z}^+$ is the number of data points. In order to simplify our problem a bit (i.e. when applying least squares), from now on we will define $M = \mathbb{R}^n$. We will disregard the bounds 0 and f'_{max} as they have little impact on the solution and the computations are much simpler without them.

Much of what we are dealing with is uncertainty: an inexact model (1.4), inexact data ($\mathbf{k}$), and some a priori information (such as a educated guess with some level of confidence). Now that we have our model and data spaces, we will turn to these uncertainties. As we have described the solution to an inverse problem as a transformation of some a priori distribution, we will begin with that prior distribution.

A priori information (or, as we will use interchangeably, prior information) is information that is obtained independently of the results of measurements [9]. Often, this will be some guess of the distribution of model parameters. We will denote this probability density by

$$\rho_M(\mathbf{m}).$$

We will choose our a priori distribution to be Gaussian, centered around a given mean $\mathbf{m}_{\text{prior}} \in M$ with covariance matrix $\mathbf{C}_{\text{M}}$. Thus our a priori distribution is

$$\rho_M(\mathbf{m}) = k_M \exp\left(-\frac{1}{2}(\mathbf{m} - \mathbf{m}_{\text{prior}})^T \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}})\right) \tag{1.7}$$

with normalization constant k_M. As we will move from this prior distribution to the posterior distribution through our knowledge of the observed data and model, we must have distributions that describe those two aspects as well.

Just as our prior information is more than a single guess, our observed data could be more or less accurate. Any act of measurement has an error involved, and this can be the result of actual measurement error or inaccuracies from the model used to determine that data. With that said, our observed data is actually described by a probability distribution $\rho_D(\mathbf{k} \,|\, \mathbf{k}_{\text{obs}})$ with $\mathbf{k}_{\text{obs}}$, the observed data, as a parameter. We will assume this distribution is Gaussian with covariance matrix $\mathbf{C}_{\text{D}}$,

$$\rho_D(\mathbf{k} \,|\, \mathbf{k}_{\text{obs}}) = k_D \exp\left(-\frac{1}{2}(\mathbf{k} - \mathbf{k}_{\text{obs}})^T \mathbf{C}_{\text{D}}^{-1}(\mathbf{k} - \mathbf{k}_{\text{obs}})\right) \tag{1.8}$$

with normalization constant k_D. As our prior distribution is independent of our observed data, the two probability density functions (1.7) and (1.8) are independent. We can then easily define their joint probability density as

$$\rho(\mathbf{k}, \mathbf{m}) = \rho_D(\mathbf{k})\rho_M(\mathbf{m}). \tag{1.9}$$

So we have two pieces of the puzzle necessary to determine the a posteriori distribution: the a priori distribution and the distribution of data. We lastly need to consider the uncertainty of the forward model $\mathbf{g}(\mathbf{m}) = \mathbf{k}$. Our (quite simple) model describes the amount of capital of a region as a function of the production function; it is inexact. The authors of [3] indeed state that a better, and likely more complex, model is needed. Therefore, the correlation between model parameters $\mathbf{m}$ and measurements of capital $\mathbf{k}$ is represented by a probability density, denoted by

$$\Theta(\mathbf{k}, \mathbf{m}). \tag{1.10}$$

This density accounts for the inexactness of the model in describing how the model parameters and outcomes are connected.

Our posterior distribution is a combination of the distributions $\rho(\mathbf{k}, \mathbf{m})$ and $\Theta(\mathbf{k}, \mathbf{m})$, i.e. combining "states of information". This combination is not simply a normalized multiplication of the two distributions. We need to define one more probability distribution on $D \times M$ before we can combine the information we have.

Consider a space X and suppose a group G acts transitively on X. A *homogeneous probability distribution* on X is a probability distribution P, with corresponding probability density function μ, that is invariant under the action of G. This is $P(G \cdot A) = P(A)$ for all measurable $A \subseteq X$.

As an example, suppose $G = X = \mathbb{R}^+$ where G has the group of operation multiplication. If we choose $\mu(x) = \frac{k}{x}$ (nonnormalizable, $k > 0$), then for any interval $A = [a, b] \in \mathbb{R}^+$ we have

$$P(A) = \int_a^b \frac{k}{x}\, dx = k \log(b/a) \quad \text{and} \quad P(\alpha A) = \int_{\alpha a}^{\alpha b} \frac{k}{x}\, dx = k \log(b/a),$$

and thus μ is a homogeneous probability density function.

As a second example, let $X = \mathbb{R}^n$ and suppose X acts on itself by component-wise addition. Observe for measurable $A \subseteq X$, if we choose $\mu(x) = k$ (again $k > 0$),

$$P(A) = \int_A k\, d\mathbf{x} = k \cdot m(A) \quad \text{and} \quad P(\mathbf{v} + A) = \int_{\mathbf{v}+A} k\, dx = k \cdot m(A)$$

where $m(A)$ is the measure of A. Note for this example and the previous one $\mu(x)$ cannot be normalized; in both cases $P(X)$ diverges. In general, this does not cause any problems [9]; we can use μ to determine relative probabilities of events.

Let $\mu_M(\mathbf{m})$ and $\mu_D(\mathbf{k})$ denote the homogeneous density functions of our model space M and data space D respectively. As $\mu_M(\mathbf{m})$ and $\mu_D(\mathbf{k})$ are independent, we can define a joint homogeneous probability as

$$\mu(\mathbf{k}, \mathbf{m}) = \mu_D(\mathbf{k})\mu_M(\mathbf{m}). \tag{1.11}$$

However, we can say more than that since we have defined our data space $D = \mathbb{R}^p$ and model space $M = \mathbb{R}^n$. With Cartesian coordinates on both spaces, we choose our homogeneous distributions to be invariant under translations as in the second example above. Therefore we choose the homogeneous distributions for both spaces to be constant: $\mu_D(\mathbf{k}) = k_d$ and $\mu_M(\mathbf{m}) = k_m$. If we had no prior information about our model parameters, we could select the homogeneous distribution as the a priori distribution [9], i.e. $\rho_M = \mu_M$, but we have chosen a Gaussian distribution for the prior ρ_M in (1.7). We can then simplify (1.11) as

$$\mu(\mathbf{k}, \mathbf{m}) = k_d k_m. \tag{1.12}$$

This homogeneous distribution plays an important role as we combine the probability distributions $\rho(\mathbf{k}, \mathbf{m})$ and $\Theta(\mathbf{k}, \mathbf{m})$.

1.6 Formulation of posterior distribution

We shall finally address the combination of the states of information describing our prior knowledge and understanding of the physical theory. In order to do this, we need to define the conjunction of two probability densities. Let $\mu(\mathbf{x})$ be a homogeneous density and suppose f_1, f_2 are two probability densities on a space X. Following [9], we define the *conjunction* of f_1 and f_2 as

$$(f_1 \wedge f_2)(\mathbf{x}) = k\frac{f_1(\mathbf{x})f_2(\mathbf{x})}{\mu(\mathbf{x})},$$

where k is a normalization constant: $k = 1/\int_X \frac{f_1(\mathbf{x})f_2(\mathbf{x})}{\mu(\mathbf{x})}d\mathbf{x}$. Note this is not simply the multiplication of f_1 and f_2. In particular, if $f_2 = \mu$, then $f_1 \wedge \mu = f_1$. In other words, the conjunction operation acts neutrally on the homogeneous density, and μ plays the role of representing no information. It is this property that we wish to preserve when we merge our prior information, model uncertainties, and data to determine the model parameters.

For our problem, the joint posterior distribution $\sigma(\mathbf{k}, \mathbf{m})$ is formed by the conjunction of $\rho(\mathbf{k}, \mathbf{m})$ (prior knowledge and measured data) and $\Theta(\mathbf{k}, \mathbf{m})$ (model uncertainties):

$$\sigma(\mathbf{k}, \mathbf{m}) = k\frac{\rho(\mathbf{k}, \mathbf{m})\Theta(\mathbf{k}, \mathbf{m})}{\mu(\mathbf{k}, \mathbf{m})}, \tag{1.13}$$

where k is the normalization constant.

Now returning to the probability density $\Theta(\mathbf{k}, \mathbf{m})$ describing the inexactness of our model in predicting the data, the marginal distribution on the model space is [9]

$$\int_D \Theta(\mathbf{k}, \mathbf{m})d\mathbf{k} = \mu_M(\mathbf{m}), \tag{1.14}$$

and hence we can define the conditional probability [7]

$$\theta(\mathbf{k} \,|\, \mathbf{m}) = \frac{\Theta(\mathbf{k}, \mathbf{m})}{\mu_M(\mathbf{m})}. \tag{1.15}$$

The density $\theta(\mathbf{k} \,|\, \mathbf{m})$ describes, given model parameters $\mathbf{m}$, a probability distribution for $\mathbf{k}$ found from the forward model, instead of one exactly predicted value. From equation (1.15), we obtain

$$\Theta(\mathbf{k}, \mathbf{m}) = \theta(\mathbf{k} \,|\, \mathbf{m})\mu_M(\mathbf{m}) = k_m\theta(\mathbf{k} \,|\, \mathbf{m}). \tag{1.16}$$

By substituting in equations (1.9), (1.12), and (1.16), we obtain

$$\sigma(\mathbf{k},\mathbf{m}) = k\frac{\rho_D(\mathbf{k})\rho_M(\mathbf{m})k_m\theta(\mathbf{k}|\mathbf{m})}{k_d k_m}$$

$$= \frac{k}{k_d}\rho_D(\mathbf{k})\rho_M(\mathbf{m})\theta(\mathbf{k}|\mathbf{m}) \tag{1.17}$$

Let $k = k/k_d$. In order to find the posterior distribution of model parameters $\sigma_M(\mathbf{m})$ (our solution), which is the marginal probability density of $\sigma(\mathbf{k},\mathbf{m})$, we need to integrate over the data space:

$$\begin{aligned}\sigma_M(\mathbf{m}) &= \int_D \sigma(\mathbf{k},\mathbf{m})d\mathbf{k} \\ &= k\int_D \rho_D(\mathbf{k})\rho_M(\mathbf{m})\theta(\mathbf{k}|\mathbf{m})d\mathbf{k} \\ &= k\rho_M(\mathbf{m})\int_D \rho_D(\mathbf{k})\theta(\mathbf{k}|\mathbf{m})d\mathbf{k}. \end{aligned} \tag{1.18}$$

Recall, we have explicit formulas for ρ_M and ρ_D in (1.7) and (1.8) respectively, but we have no description of $\theta(\mathbf{k}\,|\,\mathbf{m})$, a measure of how well the model models the real world correlation between given parameters $\mathbf{m}$ and predicted data. If the uncertainties of the model describing the theory are negligible compared to $\rho(\mathbf{k},\mathbf{m})$ (a priori and data distribution), then following [9] we let

$$\theta(\mathbf{k}|\mathbf{m}) = \boldsymbol{\delta}(\mathbf{k} - g(\mathbf{m})) \tag{1.19}$$

where $\boldsymbol{\delta}$ is the delta distribution, i.e. $\boldsymbol{\delta}(\mathbf{k}-g(\mathbf{m})) = 0$ if $\mathbf{k} \neq g(\mathbf{m})$ and $\int \boldsymbol{\delta}(\mathbf{k}-g(\mathbf{m}))d\mathbf{k} = 1$. This is an unwarranted assumption for our model, but this does correspond to one of the assumptions we make in an attempt to mimic the original paper [3]. When substituting this into our solution $\sigma_M(\mathbf{m})$, we see that

$$\sigma_M(\mathbf{m}) = k\rho_M(\mathbf{m})\int_D \rho_D(\mathbf{k})\boldsymbol{\delta}(\mathbf{k} - g(\mathbf{m}))d\mathbf{k}$$

$$= k\rho_M(\mathbf{m})\rho_D(g(\mathbf{m})). \tag{1.20}$$

By finally substituting in our a priori distribution and the distribution describing the data measurements (1.7) and (1.8), and letting $c = k \cdot k_M \cdot k_D$, we obtain

$$\begin{aligned}\sigma_M(\mathbf{m}) &= c \cdot \exp\left(-\frac{1}{2}\left[(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})^T \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})\right.\right. \\ &\qquad \left.\left. + (\mathbf{m} - \mathbf{m}_{\text{prior}})^T \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}})\right]\right) \\ &= c \cdot \exp\left(-\frac{1}{2}\left[\|\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}\|_D^2 + \|\mathbf{m} - \mathbf{m}_{\text{prior}}\|_M^2\right]\right). \qquad (1.21)\end{aligned}$$

Therefore, under the assumptions that our a priori and data distributions are Gaussian, our data space D is linear, and the unrealistic assumption that our model is exact, the solution to our inverse problem is given by $\sigma_M(\mathbf{m})$ in (1.21). The remainder of this paper develops the two methods, Metropolis and least squares minimization, used to estimate properties of this solution. First however, as both Metropolis and least squares depend on the solution to the forward problem $\mathbf{g}(\mathbf{m})$, our first task is to solve the forward problem.

Chapter 2

Forward Problem

In the resolution of an inverse problem, in this case describing the distribution of the slopes $\mathbf{m}$ of the piecewise linear function $\tilde{f}$ based on the amount of capital $\mathbf{k}$, the solution to the forward problem $\mathbf{g}(\mathbf{m}) = \mathbf{k}$ is paramount. Indeed, our posterior distribution (1.21) depends greatly on $\mathbf{g}(\mathbf{m})$. As we shall see in subsequent chapters, the two methods we use to describe aspects of the posterior distribution require the solution of the forward problem many times for various inputs $\mathbf{m}$. We will solve the forward problem using a finite difference scheme.

2.1 The problem

Recall our model from section 1.2:

$$\begin{cases} \dfrac{\partial}{\partial t}k(x,t) = \dfrac{\partial^2}{\partial x^2}k(x,t) + a(x,t)f(k(x,t)) - \delta k(x,t) \\ k_x(0,t) = 0, \quad k_x(L,t) = 0 \end{cases} \tag{2.1}$$

where $x \in [0, L]$. Also recall a is the technology level and δ is the depreciation. The continuous forward problem of this model is to predict the capital stock of households at position x and time t, i.e. $k(x,t)$, with a given production function f. This solution is $\mathbf{g}(f) = k$, which involves solving the partial differential equation for some function f.

As we are discretizing the forward problem, now $\mathbf{g}(\mathbf{m})$, instead of solving (2.1) for a given $f \in \mathcal{F}$, we will solve (2.1) where $f = \tilde{f}$, a piecewise linear function. So, the solution to our forward problem $\mathbf{g}(\mathbf{m})$ is the solution to

$$\begin{cases} k_t = k_{xx} + a\tilde{f}(k) - \delta k \\ k_x(0,t) = k_x(L,t) = 0 \end{cases} \tag{2.2}$$

where $\tilde{f}$ is determined by $\mathbf{m}$. This amounts to numerically solving a partial differential equation (PDE), which again we accomplish using finite differences.

2.2 Finite difference discretization

Consider the domain $[0, L] \times [0, T]$ with some given technology level $a(x, t)$ and depreciation δ. To solve equation (2.2), we will use a finite difference scheme. We will discretize the spatial derivatives using the classic centered difference scheme (method of lines), and we will then tackle the time derivatives. Solving this stiff PDE [6] using just an explicit time step (forward Euler) would require a prohibitively small time step for the scheme to be stable. Using an implicit time step would allow for a larger time step, but the PDE is nonlinear, so we would have to solve a nonlinear system of equations at each of those larger time steps (e.g. using Newton's method). However, we can use an implicit-explicit method [1],[6], which allows us to solve this PDE using backward Euler for the linear terms $k_{xx}(x, t) - \delta k(x, t)$ and forward Euler for the nonlinear term $a(x, t)f(k(x, t))$. By using this approach we will not be subject to the unreasonably short time steps necessary for forward Euler alone.

To set up the implicit-explicit scheme, we will begin by discretizing space and time. For this section, we will denote m as the number of spatial grid points (not including the boundary) and $n + 1$ as the number of temporal grid points (not including initial time). By choosing space step Δx and time step Δt, we put a grid on the domain of k where $\Delta x = \frac{L}{m+1}$ and $\Delta t = \frac{T}{n+1}$, and we denote $x_0 = 0$, $x_{m+1} = L$, $t_0 = 0$ and $t_{n+1} = T$.

Applying the centered difference approximation for k_{xx}, we have

$$k_{xx}(x_i, t_j) \approx \frac{k(x_{i-1}, t_j) - 2k(x_i, t_j) + k(x_{i+1}, t_j)}{(\Delta x)^2}. \tag{2.3}$$

Now consider the backward difference approximation for k_t,

$$k_t(x_i, t_{j+1}) \approx \frac{k(x_i, t_{j+1}) - k(x_i, t_j)}{\Delta t}, \tag{2.4}$$

and the forward difference approximation

$$k_t(x_i, t_j) \approx \frac{k(x_i, t_{j+1}) - k(x_i, t_j)}{\Delta t}. \tag{2.5}$$

As a simple implicit-explicit scheme, we apply backward Euler to the linear terms $k_{xx} - \delta k$ and forward Euler to the nonlinear term $af(k)$. The discretized PDE in (2.2) becomes

$$\frac{k(x_i, t_{j+1}) - k(x_i, t_j)}{\Delta t} = \frac{k(x_{i-1}, t_{j+1}) - 2k(x_i, t_{j+1}) + k(x_{i+1}, t_{j+1})}{(\Delta x)^2} + a(x_i, t_j) f(k(x_i, t_j)) - \delta k(x, t_{j+1}) + (\text{truncation error}). \tag{2.6}$$

Now we denote by $K_{i,j}$ our approximation to $k(x_i, t_j)$ for $i \in \{0, 1, .., m+1\}$ and $j \in \{0, 1, ..., n+1\}$. By using this notation, (2.6) becomes the iterative system

$$\frac{K_{i,j+1} - K_{i,j}}{\Delta t} = \frac{K_{i-1,j+1} - 2K_{i,j+1} + K_{i+1,j}}{(\Delta x)^2} + a_{ij} f(K_{i,j}) - \delta K_{i,j+1}. \tag{2.7}$$

It is perfectly reasonable to solve this system for generic values of i, but when $i = 0$ or L, the values x_{i-1} and x_{i+i} respectively lie outside of the domain of k. For these locations, the boundary condition $k_x(0, t) = k_x(L, t) = 0$ must be employed. Similar to the centered scheme for k_{xx}, we can use a finite centered difference scheme to discretize the boundary conditions:

$$k_x(0, t) \approx \frac{k(\Delta x, t) - k(-\Delta x, t)}{2\Delta x} = 0.$$

Thus $k(\Delta x, t) = k(-\Delta x, t)$ which, by using our discrete notation, is $K_{-1,j} = K_{1,j}$. By a similar computation, $K_{L+1,j} = K_{L-1,j}$. Let $\mu = \Delta t/(\Delta x)^2$. Denote by $K^{(j)}$ the vector representing capital assets at time j.

By isolating $K_{i,j}$ to one side and setting up (2.7) with boundary conditions in matrix notation, we obtain the sparse tridiagonal system

$$AK^{(j+1)} = K^{(j)} - \Delta t a^{(j)} f(K^{(j)}) \tag{2.8}$$

where

$$A = \begin{pmatrix} 1 + 2\mu + \delta\Delta t & -2\mu & & & & \\ -\mu & 1 + 2\mu + \delta\Delta t & -\mu & & & \\ & -\mu & 1 + 2\mu + \delta\Delta t & -\mu & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\mu & 1 + 2\mu + \delta\Delta t & -\mu \\ & & & & -2\mu & 1 + 2\mu + \delta\Delta t \end{pmatrix}.$$

Note the -2μ in the first and last rows are enforcing the boundary conditions. By solving the system (2.8) at each time step, we will find the numerical solution K. When we vectorize we obtain $\mathbf{k}$.

2.3 Implementation and solution

Following the assumptions in [3], let $L = 50$ and $T = 150$. We can then specify our mesh by choosing $\Delta x = 0.2$ and $\Delta t = 0.6$. This gives us a 251×251 point grid, with $x_0 = 0$, $x_{251} = 50$, $t_0 = 0$, and $t_{251} = 150$. Furthermore, suppose $a(x,t) \equiv 1$ and $\delta = 0.05$. We will test our solution to the forward problem by choosing a production function $f \in \mathcal{F}$, namely

$$f(k) = \frac{0.0005k^4}{1 + 0.0005k^4}. \tag{2.9}$$

This function is plotted in Figure 1.2. For two different piecewise linear choices of an initial distribution of capital, $k_1(x,0)$ and $k_2(x,0)$, we obtain solutions $k_1(x,t)$ and $k_2(x,t)$ in Figure 2.1.

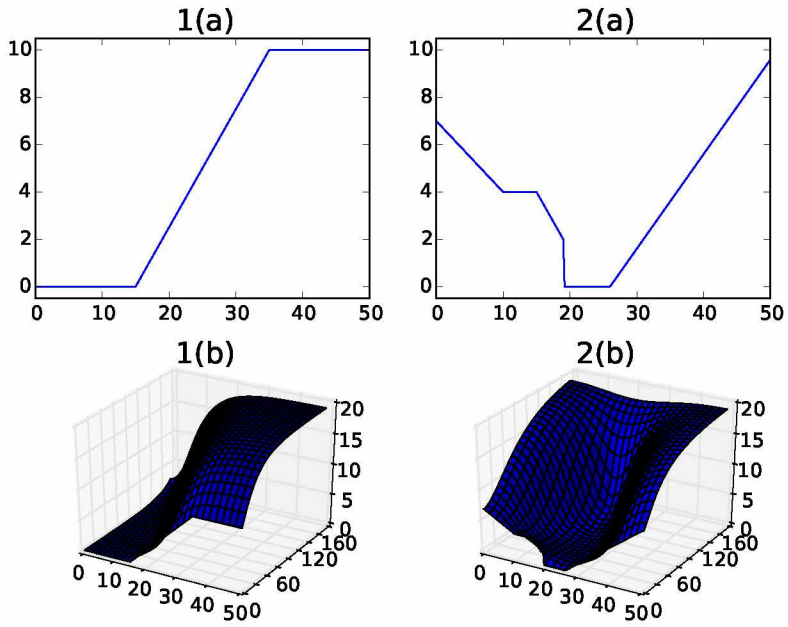

Figure 2.1: 1(a) $k_1(x,0)$, 2(a) $k_2(x,0)$, 1(b) Solution k_1 from production function f, 2(b) Solution k_2 from production function f

2.4 Verification

As the solution of the forward problem is one step in our ultimate goal of solving the inverse problem, we need to verify that the forward problem is being solved correctly. That is, we need to show the numerical solution produced by the implementation of this scheme converges to the actual (exact) solution. Of course equation (2.1) can be very difficult to solve exactly (if even possible) for some function f. Therefore, we will manufacture a function f for which we know the exact solution k for (2.1). Consider $\hat{k}(x,t) = \left(\cos^2 \frac{\pi x}{L} - \frac{1}{2}\right) e^{.01t}$ where $x \in [0, L]$ and $t \in [0, T]$. Note this function satisfies the homogeneous Neumann boundary conditions. Also, $\hat{k}(x,t) < 0$ for some (x,t) which would denote negative values of capital. But, as we are just verifying the implementation of the scheme that solves the PDE numerically, k need not be positive. We obtain the partial derivatives

$$\begin{aligned} \hat{k}_{xx} &= -\frac{2\pi^2}{L^2}\left(\cos^2 \frac{\pi x}{L} - \sin^2 \frac{\pi x}{L}\right) e^{.01t} \\ &= -\frac{4\pi^2}{L^2}\left(\cos^2 \frac{\pi x}{L} - \frac{1}{2}\right) e^{.01t} \qquad (2.10) \\ \hat{k}_t &= \frac{1}{100}\left(\cos^2 \frac{\pi x}{L} - \frac{1}{2}\right) e^{.01t}. \qquad (2.11) \end{aligned}$$

By substituting these partial derivatives into (2.1), we find

$$\begin{aligned} \hat{f}(\hat{k}) &= \hat{k}_t - \hat{k}_{xx} + \delta\hat{k} \\ &= \left(\frac{1}{100} + \frac{4\pi^2}{L^2} + \delta\right)\hat{k}. \qquad (2.12) \end{aligned}$$

Therefore $\hat{k}$ is the exact solution to $k_t = k_{xx} + \hat{f}(k) - \delta k$. Let k_{test} be the numerical solution to the forward problem (2.1) we computed when $f = \hat{f}$. For $L = 50$ and $\delta = 0.05$, we can compare $\hat{k}$ evaluated at grid points to k_{test}. Consider the refinement path $h = \Delta x = \frac{1}{3}\Delta t$. When we take $h \to 0$ we see $\|\hat{k} - k_{\text{test}}\|_\infty \to 0$. Furthermore, $\|\hat{k}^h - k^h_{\text{test}}\|_\infty = O(h^1)$ (see Figure 2.2). This is the predicted rate as we used a first-order accurate scheme based on forward and backward Euler [6].

Now that we have a method to solve the forward problem, we will move on to the methods used to solve the inverse problem.

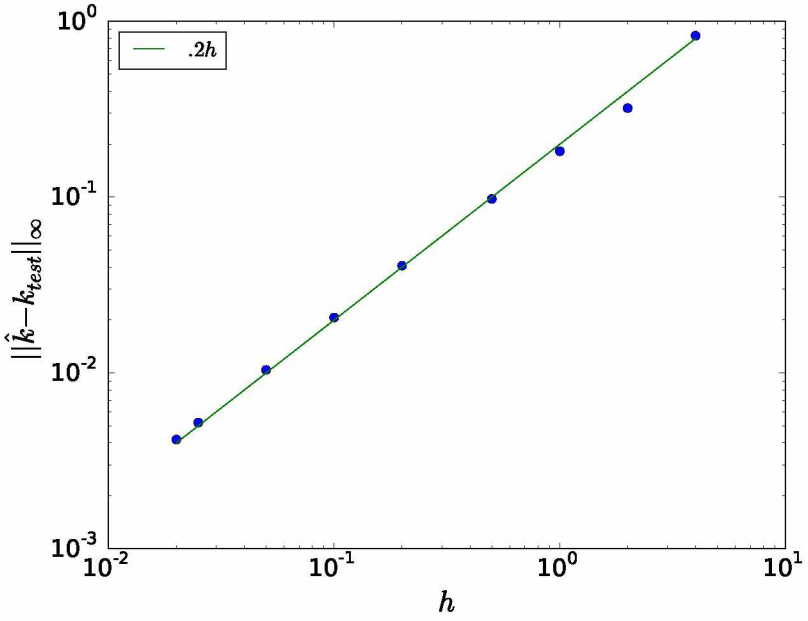

Figure 2.2: Plot showing error of numerical solution as a function of spatial grid spacing

Chapter 3

Sampling of Parameter Space

As we have stated previously, the solution to an inverse problem is generally a probability distribution over the model space [9]. For complex posterior distributions (e.g. numerous local maxima), describing aspects of this solution may only be possible by sampling the distribution. Recall we have formulated the posterior distribution as

$$\sigma_M(\mathbf{m}) = c \cdot \exp\left(-\frac{1}{2}\left[(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})^t \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^t \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}})\right]\right).$$

We will use a robust approach (that would work on more complex problems) to sample this distribution. We will then compare some characteristics of this sample with the computation of the inverse problem solution through least squares optimization (Chapter 4).

3.1 Markov chain Monte Carlo approach

One way to sample this distribution, finding a collection of points $\{\mathbf{m}^{(i)}\}_{i=1}^{K}$ where $K \in \mathbb{Z}^+$, is using random techniques, such as Markov chain Monte Carlo. The essence of this approach is to generate samples of points through a random walk of the space in which each step is dependent only on the present location.

If we begin at some sample point $\mathbf{m}^{(i)}$, we wish to define rules that select the next point in our sample $\mathbf{m}^{(i+1)}$. As we will use a Markov chain approach, this move will only depend on our current location $\mathbf{m}^{(i)}$, and no previous location $\mathbf{m}^{(i-k)}$ for $k = 1, 2, 3, \ldots$. There are four aspects of this process: a method of proposing a step, a rule to accept that move (together which define a random walk), a decision about which iteration to start sampling, and a decision on how many samples to include. The locations near $\mathbf{m}_{\text{prior}}$ may not be good samples of the posterior distribution; we may need to make many steps (burn in process) before we are sampling from the posterior distribution. Each sample will be highly correlated to the previous one, and thus the number of samples will determine how well we have sampled

the posterior distribution. As this may turn out to be very large, we will not include every sample in our collection.

We will begin by defining how to take a proposed step from point $\mathbf{m}^{(i)}$ to $\mathbf{m}^*$. Note, $\mathbf{m}^*$ is not automatically admitted to our sample; this would not take into account our distribution on the model space M, only its constant homogeneous distribution. Given a point $\mathbf{m}$ with components m_j, we define a random step to $\mathbf{m}^*$ as

$$m_j^* = m_j^{(i)} + \varepsilon_j, \tag{3.1}$$

where ε_j is a single sample point from a normal distribution $N(0, \sigma)$. Note each component of $\mathbf{m}$ is perturbed and σ will determine the size of the step. If the step is too small, not enough of the model space will be explored but if the step is too large, the Metropolis algorithm (introduced next to determine the success of the proposed step) will reject many steps leading to an inefficient algorithm [9]. The size of the step will be crucial. Once that proposed step is made, we have to determine if we stay there and if that point is then included in the sample.

3.2 Metropolis implementation

Now that we have a method of proposing a step in a random walk, we need to determine if that step will be taken. We will use the Metropolis algorithm to accept or reject our move. Suppose we have a proposed transition from $\mathbf{m}^{(i)}$ to $\mathbf{m}^*$. From [9], we will accept the move to $\mathbf{m}^*$ if

$$(1) \quad \sigma_M(\mathbf{m}^*) \geq \sigma_M(\mathbf{m}^{(i)}), \tag{3.2}$$

or we will accept the move from $\mathbf{m}^{(i)}$ to $\mathbf{m}^*$ with probability $\sigma_M(\mathbf{m}^*)/\sigma_M(\mathbf{m}^{(i)})$ if

$$(2) \quad \sigma_M(\mathbf{m}^*) < \sigma_M(\mathbf{m}^{(i)}). \tag{3.3}$$

If $\mathbf{m}^*$ is accepted, then $\mathbf{m}^{(i+1)} = \mathbf{m}^*$. On the other hand if $\mathbf{m}^*$ is rejected, then $\mathbf{m}^{(i+1)} = \mathbf{m}^{(i)}$. We can run the Metropolis algorithm repeatedly. This will define the random walk, but we will not include every point in our collection. Along with the starting point and sample size, we will determine with what frequency to include accepted moves as samples. This selection is primarily needed to prevent memory problems that would arise from storing a sufficient number of samples. Ultimately, this sample size will be determined by the problem and progression of the Metropolis algorithm. These parameters will be defined in the results chapter.

We will attempt to compare the mean and credible (confidence) intervals of each component of the sample obtained through Metropolis with the maximum a posteriori estimate (or posterior mode) $\mathbf{m}_{\text{MAP}}$ found through least squares and covariance matrix $\tilde{\mathbf{C}}_{\text{M}}$ of $\sigma_M(\mathbf{m})$ when we linearize at that point. We consider the least squares algorithm next.

Chapter 4

Least Squares

Monte Carlo techniques may be necessary to describe the posterior distribution of an inverse problem; the distribution could have any number of local maximums or other features that can only be found through sampling the distribution. This depends on the nature of $\mathbf{g}$ [9]. We do not know the complexity of our posterior distribution, but we can go ahead and determine where the distribution is maximized by minimizing the sum of squared residuals, i.e. least squares. We will recall our posterior distribution

$$\sigma_M(\mathbf{m}) = c \cdot \exp\left(-\frac{1}{2}\left[(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})^t \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^t \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}})\right]\right). \tag{4.1}$$

Observe when $\mathbf{C}_{\text{M}}^{-1} = \frac{1}{\sigma_1^2} I$ and $\mathbf{C}_{\text{D}}^{-1} = \frac{1}{\sigma_2^2} I$, where σ_1 and σ_2 are respective standard deviations for $\rho_M(\mathbf{m})$ and $\rho_D(\mathbf{g}(\mathbf{m}))$, the exponential is a sum of squares. We will determine the maximum a posteriori estimate $\mathbf{m}_{\text{MAP}}$ of this distribution by finding the maximum value of $\sigma_M(\mathbf{m})$. We will then linearize $\mathbf{g}$ at this point, and in a neighborhood of $\mathbf{m}_{\text{MAP}}$ the posterior distribution will be approximately Gaussian [9]. We can then determine the covariance matrix, which we will denote $\tilde{\mathbf{C}}_{\text{M}}$. By finding the maximum a posteriori estimate, we are essentially applying Tikhonov regularization, with the regularization term coming from our a priori distribution. However, we are going one step further to attempt to answer how certain we are of the maximum a posteriori estimate.

4.1 Misfit function

The formulation of the posterior distribution $\sigma_M(\mathbf{m})$ in (4.1) gives rise to the *misfit function* $S(\mathbf{m})$ where

$$2S(\mathbf{m}) = (\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})^t \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^t \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}}). \tag{4.2}$$

We note $\sigma_M(\mathbf{m}) = c \cdot \exp(-S(\mathbf{m}))$. The misfit function is a measure of how poorly the data and our prior knowledge fits some input model parameter $\mathbf{m}$. We wish to find the posterior mode $\mathbf{m}_{\text{MAP}}$ which will be where $\sigma_M(\mathbf{m})$ is maximized. Note, this goal is equivalent to minimizing $S(\mathbf{m})$. An effective method for minimizing $S(\mathbf{m})$ is to compute the gradient $\nabla S(\mathbf{m})$, whose ith component is $\frac{\partial S}{\partial m_i}$, where $\mathbf{m} = (m_1, m_2, ..., m_n)$. We then need to determine where that gradient is 0. This is hard (if not impossible) to do exactly, especially as the dimension of the problem gets large. We must use a method of numerical optimization.

4.2 Gauss-Newton method

We will employ Newton's method to solve the equation $\nabla S(\mathbf{m}) = 0$. From Taylor's Theorem, we have

$$\nabla S\left(\mathbf{m}^{(1)}\right) = \nabla S\left(\mathbf{m}^{(0)}\right) + \nabla^2 S\left(\mathbf{m}^{(0)}\right) \Delta\mathbf{m} + O\left(\Delta\mathbf{m}^2\right),$$

where $\Delta\mathbf{m} = \mathbf{m}^{(1)} - \mathbf{m}^{(0)}$. Let $\mathbf{s} = \Delta\mathbf{m}$. By dropping higher order terms of $\mathbf{s}$ and setting the gradient at $\mathbf{m}^{(1)}$ equal to 0, we obtain

$$\nabla^2 S\left(\mathbf{m}^{(0)}\right)\mathbf{s} = -\nabla S\left(\mathbf{m}^{(0)}\right). \tag{4.3}$$

Note $\nabla^2 S$ is the Hessian of S, i.e., the matrix of second derivatives of S. By starting with an initial guess $\mathbf{m}^{(0)}$, this gives us the Newton update

$$\mathbf{m}^{(n+1)} = \mathbf{m}^{(n)} + \alpha\mathbf{s} \tag{4.4}$$

where we solve $\nabla^2 S\mathbf{s} = -\nabla S$ for $\mathbf{s}$ and $\alpha > 0$ is a scalar parameter. We call $\mathbf{s}$ the descent direction, and when $\alpha = 1$, this is the standard Newton method. By starting with an initial guess ($\mathbf{m}_{\text{prior}}$ is not unreasonable), computing subsequent iterates will hopefully converge to the minimizer of $S(\mathbf{m})$. Applying Newton's method requires us to compute the gradient and Hessian of $S(\mathbf{m})$. The gradient of $S(\mathbf{m})$ is

$$\nabla S = \frac{\partial S}{\partial \mathbf{m}} = \begin{pmatrix} \frac{\partial S}{\partial m_1} \\ \vdots \\ \frac{\partial S}{\partial m_n} \end{pmatrix}.$$

We can compute each component of the gradient at point $\mathbf{m}$ using the product rule as follows:

$$\frac{\partial S}{\partial m_j}(\mathbf{m}) = \left(\frac{\partial \mathbf{g}}{\partial m_j}(\mathbf{m})\right)^T \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}) + e_j^T \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}}) \tag{4.5}$$

where we approximate $\frac{\partial g_i}{\partial m_j}$ in the next section. Observe $\frac{\partial \mathbf{g}}{\partial \mathbf{m}}$ is the matrix of partial derivatives, i.e. the Jacobian. If we denote this as $\mathbf{G}$, the gradient at $\mathbf{m}$ is

$$\nabla S(\mathbf{m}) = (\mathbf{G}(\mathbf{m}))^T \mathbf{C}_{\mathrm{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\mathrm{obs}}) + \mathbf{C}_{\mathrm{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\mathrm{prior}}). \tag{4.6}$$

As we have a way to compute the gradient, ∇S, to find the descent direction for the Gauss-Newton method, we just need the Hessian of S (evaluated at the point $\mathbf{m}$). Note the Hessian is the $n \times n$ matrix

$$\nabla^2 S = \frac{\partial^2 S}{\partial \mathbf{m}^2}$$

where the ijth entry of $\nabla^2 S$ is $(\nabla^2 S)_{ij}$. We can compute these entries by differentiating (4.5). Applying the product rule again, we have

$$\left(\nabla^2 S\right)_{ij}(\mathbf{m}) = \left(\frac{\partial \mathbf{g}}{\partial m_i}(\mathbf{m})\right)^T \mathbf{C}_{\mathrm{D}}^{-1} \left(\frac{\partial \mathbf{g}}{\partial m_j}(\mathbf{m})\right) + e_i^T \mathbf{C}_{\mathrm{M}}^{-1} e_j + \frac{\partial}{\partial m_j}\left(\frac{\partial \mathbf{g}}{\partial m_i}(\mathbf{m})\right)^T \mathbf{C}_{\mathrm{D}}^{-1}\left(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\mathrm{obs}}\right). \tag{4.7}$$

Now, each step of Newton will result in a decrease of $S(\mathbf{m})$ as long as a descent direction is chosen - it need not be in the *steepest* descent direction. If either the residual $(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\mathrm{obs}})$ is small or the nonlinearity of $\mathbf{g}(\mathbf{m})$ is small (causing the second derivative of $\mathbf{g}$ to be small), then the last term of (4.7) will be small [9]. We will proceed by dropping the last term of the Hessian [8], and use the approximation of the Hessian $\tilde{\nabla}^2 S$ with entries

$$\left(\tilde{\nabla} S\right)_{ij}(\mathbf{m}) = \left(\frac{\partial \mathbf{g}}{\partial m_i}(\mathbf{m})\right)^T \mathbf{C}_{\mathrm{D}}^{-1} \left(\frac{\partial \mathbf{g}}{\partial m_j}(\mathbf{m})\right) + (\mathbf{C}_{\mathrm{M}}^{-1})_{ij}. \tag{4.8}$$

Using $\mathbf{G}$ to denote the Jacobian as we did with the gradient, the approximate Hessian at the point $\mathbf{m}$ is

$$\tilde{\nabla}^2 S(\mathbf{m}) = (\mathbf{G}(\mathbf{m}))^T \mathbf{C}_{\mathrm{D}}^{-1}(\mathbf{G}(\mathbf{m})) + \mathbf{C}_{\mathrm{M}}^{-1}. \tag{4.9}$$

By using the approximation of the Hessian $\tilde{\nabla}^2 S$ instead of $\nabla^2 S$, we will call the method described in (4.3) and (4.4) as Gauss-Newton. Since we now have a way to find the maximum a posteriori estimate and covariance of the posterior distribution, we just need to compute the gradient and Hessian.

4.3 Approximation of derivatives

As seen earlier, both the gradient ∇S and approximate Hessian $\tilde{\nabla}^2 S$ require the computation of the Jacobian $\mathbf{G}$, but only first derivatives of $\mathbf{g}(\mathbf{m})$ are needed for this. For simplicity, we will use a difference quotient to approximate these partial derivatives where the ijth

component of $\mathbf{G}$ is found by

$$\frac{\partial g_i}{\partial m_j} \approx \frac{\mathbf{g}\left(\begin{pmatrix} m_1 \\ \vdots \\ m_j + \Delta m \\ \vdots \\ m_n \end{pmatrix}\right)_i - \mathbf{g}\left(\begin{pmatrix} m_1 \\ \vdots \\ m_j - \Delta m \\ \vdots \\ m_n \end{pmatrix}\right)_i}{2\Delta m}. \tag{4.10}$$

Now recall, computing $\mathbf{g}$ depends on the piecewise linear function $\tilde{f}$ that is uniquely determined by $\mathbf{m}$. Therefore, to evaluate $\mathbf{g}$ in each term of (4.10) we must find the piecewise linear functions specified by both $\begin{pmatrix} m_1 & \dots & m_i + \Delta m & \dots & m_n \end{pmatrix}^T$ and $\begin{pmatrix} m_1 & \dots & m_i - \Delta m & \dots & m_n \end{pmatrix}^T$.

Of course, with a main factor of the gradient and Hessian being an approximation, that is the best we can hope for in the computation of ∇S and $\tilde{\nabla}^2 S$. With the approximations of both ∇S and $\tilde{\nabla}^2 S$ evaluated at a points $\mathbf{m}$, we can proceed with the Gauss-Newton method described in (4.3) and(4.4) from an initial guess $\mathbf{m}^{(0)}$. We will proceed to find a sequence of point $\mathbf{m}^{(0)}, \mathbf{m}^{(1)}, ...\mathbf{m}^{(j)}$ until $\nabla S(\mathbf{m}^{(j)})$ is sufficiently small. We can then claim $\mathbf{m}_{\text{MAP}} = \mathbf{m}^{(j)}$.

Once we find $\mathbf{m}_{\text{MAP}}$, we can linearize $\mathbf{g}(\mathbf{m})$ at that point. Dropping second order terms, this is

$$\mathbf{g}(\mathbf{m}) \approx \mathbf{g}(\mathbf{m}_{\text{MAP}}) + \mathbf{G}(\mathbf{m}_{\text{MAP}})(\mathbf{m} - \mathbf{m}_{\text{MAP}}), \tag{4.11}$$

where $\mathbf{G}(\mathbf{m}_{\text{MAP}})$ is the Jacobian evaluated at the maximum a posteriori point. We then estimate the posterior covariance [9] by

$$\tilde{\mathbf{C}}_{\text{M}} = \left[\tilde{\nabla}^2 S(\mathbf{m}_{\text{MAP}})\right]^{-1} = \left[\mathbf{G}(\mathbf{m}_{\text{MAP}})^T \mathbf{C}_{\text{D}}^{-1} \mathbf{G}(\mathbf{m}_{\text{MAP}}) + \mathbf{C}_{\text{M}}^{-1}\right]^{-1}.$$

Now that we have described the plan of attack for both approaches (Metropolis and least squares), we will implement both algorithms to describe the posterior distribution.

Chapter 5

Results

We finally get to solving the inverse problem - by both the Metropolis algorithm to sample the posterior distribution and least squares to determine the location of the maximum and covariance of the posterior distribution. To determine the distribution of parameters $\mathbf{m}$, we need to start with some capital data $\mathbf{k}_{\text{obs}}$. We do not have actual data for a one-dimensional region or country with which to work. Instead, we will generate data using our model and a chosen production function f and introduce Gaussian noise.

Recall then our solution *is* the posterior distribution

$$\sigma_M(\mathbf{m}) = c \cdot \exp\left(-\frac{1}{2}\left[(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}})^t \mathbf{C}_{\text{D}}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{k}_{\text{obs}}) + (\mathbf{m} - \mathbf{m}_{\text{prior}})^t \mathbf{C}_{\text{M}}^{-1}(\mathbf{m} - \mathbf{m}_{\text{prior}})\right]\right), \tag{5.1}$$

but this solution is not tractable. We cannot visualize $\sigma_M(\mathbf{m})$, but we can try to describe a few features. In particular, we are interested primarily in finding where it is maximized, $\mathbf{m}_{\text{MAP}}$ (which is what the authors did in [3]), mean, and also some way to quantify the certainty of those model parameters. That is our goal, and as we have two methods at our disposal, in the end we compare summary statistics found by both methods.

In particular, we will sample the posterior distribution using the Metropolis algorithm, and then compute the sample mean, which we will denote $\hat{\mathbf{m}}$, and credible intervals (the middle 95% values of each component of $\mathbf{m}$). We will also find the minimizer of $S(\mathbf{m})$, $\mathbf{m}_{\text{MAP}}$, and linearize σ_M at that point. We will then compute the covariance matrix, $\tilde{\mathbf{C}}_{\text{M}}$, of this distribution. Lastly, we show the comparison of $\hat{\mathbf{m}}$ to $\mathbf{m}_{\text{MAP}}$ and the 95% credible (confidence) intervals with the standard deviations in the diagonals of $\tilde{\mathbf{C}}_{\text{M}}$.

Now $\sigma_M(\mathbf{m})$ would be Gaussian if $\mathbf{g}$ were linear, (i.e. $\mathbf{g}(\mathbf{m}) = \mathbf{G}\mathbf{m}$) [9], and in that case we could directly compute its mean and covariance. Alas, $\mathbf{g}$ is not linear and hence $\sigma_M(\mathbf{m})$ is not Gaussian. The posterior distribution depends on how close (or far) $\mathbf{g}(\mathbf{m})$ is from being

linear. Therefore, the more nonlinear $\mathbf{g}(\mathbf{m})$ is, the less likely the linearization around $\mathbf{m}_{\text{MAP}}$ will produce a covariance $\tilde{\mathbf{C}}_{\text{M}}$ that accurately describes $\sigma_M(\mathbf{m})$ away from $\mathbf{m}_{\text{MAP}}$. Also $\hat{\mathbf{m}}$ depends on the samples generated, and two different collections of sample points will have different sample means (and covariances). Therefore, we can't expect the comparison of our results to be exact, but we do expect them to be "close". At this point, all that remains will be recalling the context of the original inverse problem.

Ultimately, we are trying to recover the production function f depending on capital data of a region. The maximum a posteriori estimate (or the sample mean) can define that piecewise function we seek, and the covariance matrix (or credible intervals) can quantify our certainty of that production function.

The results in this final chapter come from running the Metropolis algorithm and the Gauss-Newton method (which both use the scheme for solving the forward problem). These three algorithms have been described in the previous chapters. In the first section in this chapter we concretely define our distribution of data ρ_D and a priori distribution ρ_M. In the subsequent section we define the parameters necessary to fully implement our two methods for describing our a posteriori distribution σ_M. We lastly implement them both and describe our solution.

5.1 Data and a priori distributions

To describe the posterior distribution, we must first have a concrete definition of the distribution of data $\rho_D(\mathbf{k})$ and the a priori distribution $\rho_M(\mathbf{m})$. With these functions in hand, we can implement the Metropolis algorithm and Gauss-Newton both described earlier. For the solution calculated here, we have $n = 19$, so $\mathbf{m} \in \mathbb{R}^{19}$.

Let us begin with the data for capital. In addition to some given or observed capital data $\mathbf{k}_{\text{obs}}$, we must also have a description of the errors of those measurements, the covariance matrix $\mathbf{C}_{\text{D}}$. We begin by generating data by our model $\mathbf{g}(\mathbf{m}) = \mathbf{k}_{\text{obs}}$, i.e. we have solved the forward problem given slopes of some piecewise production function and initial distribution of capital. We will use the $\mathbf{m} \in \mathbb{R}^{19}$ which come from approximating the function

$$f(k) = \frac{0.0005k^4}{1 + 0.0005k^4}. \tag{5.2}$$

and initial capital

$$k(x,0) = \begin{cases} 0, & \text{for } 0 \le x < 15 \\ .5x - 7.5, & \text{for } 15 \le x < 35 \\ 10, & \text{for } 35 \le x \le 50, \end{cases} \tag{5.3}$$

which are the same functions used to produce the data in [3]. See Figure 1.2 for the plot of f.

We can compute $\mathbf{k}_{\text{obs}}$ from solving the forward problem described in section 2.2. With components k_i of $\mathbf{k}_{\text{obs}}$, we can introduce errors in our data based on our perceived level of accuracy of that data - this is represented by $\mathbf{C}_{\text{D}}$. We suppose each data point is independent and the mean of a normal distribution, and thus $\mathbf{C}_{\text{D}}$ is diagonal. Thus we define the ith diagonal entry of $\mathbf{C}_{\text{D}}$ to be

$$(\mathbf{C}_{\text{D}})_i = \begin{cases} (k_i/10)^2, & \text{if } k_i \neq 0 \\ 10^{-4}, & \text{if } k_i = 0 \end{cases} \tag{5.4}$$

where k_i is the ith component of $\mathbf{k}_{\text{obs}}$. Note we need $\mathbf{C}_{\text{D}}$ to be nonsingular, so no entries along the diagonal of the matrix can be 0. Using the standard deviations defined in (5.4), we introduce Gaussian noise into our data. The plots of the produced data (a) and perturbed data (b) are in Figure 5.1. We will use the perturbed data as $\mathbf{k}_{\text{obs}}$.

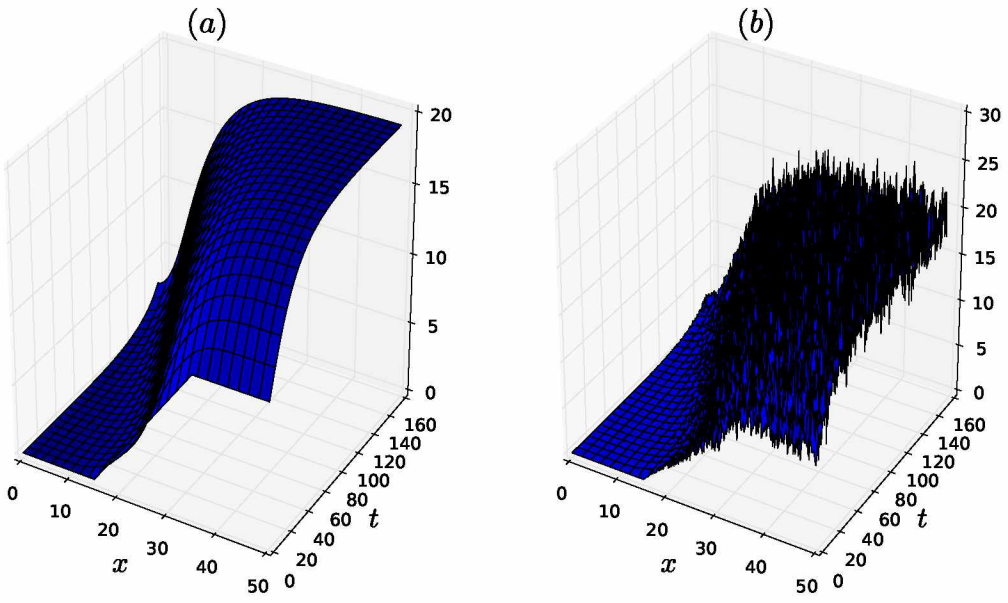

Figure 5.1: (a) Plot of produced data, (b) Plot of perturbed produced data

As we recall from (1.8), the probability distribution of our data is

$$\rho_D(\mathbf{k}) = k_D \exp\left(-\frac{1}{2}\left[(\mathbf{k}-\mathbf{k}_{\text{obs}})^T \mathbf{C}_{\text{D}}(\mathbf{k}-\mathbf{k}_{\text{obs}})\right]\right),$$

and as we have defined $\mathbf{k}_{\text{obs}}$ and $\mathbf{C}_{\text{D}}$, we can move onto our a priori distribution. Similar to the distribution of data, we have assumed our prior distribution $\rho_M(\mathbf{m})$ is Gaussian. We need to pick quantities for the mean ($\mathbf{m}_{\text{prior}}$) and covariances (entries of $\mathbf{C}_{\text{M}}$) of this distribution.

We will make an educated guess ($f_{\text{prior}} \in \mathcal{F}$) of the convex-concave production function f we are ultimately trying to recover. From there we can determine $\mathbf{m}_{\text{prior}}$ from the piecewise approximation of f_{prior} at 19 nonzero nodes. Let

$$f_{\text{prior}} = 0.9\left(\frac{0.001k^4}{1+0.001k^4}\right). \tag{5.5}$$

The graph of f_{prior} along with its piecewise linear approximation $\tilde{f}_{\text{prior}}$ and f from (5.2) is shown in Figure 5.2(a).

We can determine $\mathbf{m}_{\text{prior}}$ by computing the slopes between the nodes of $\tilde{f}_{\text{prior}}$. For simplicity, we will assume the components of $\mathbf{m}$ are independent, and thus $\mathbf{C}_{\text{M}}$ is diagonal. It may be reasonable to be more certain of small components of $\mathbf{m}$ (i.e. where f_{prior} is flatter) than larger ones (where f is steeper) so we will use relative standard deviations at each node. We define the ith diagonal entry of $\mathbf{C}_{\text{M}}$ to be

$$(\mathbf{C}_{\text{M}})_i = (0.15m_i + 0.0004)^2 \tag{5.6}$$

where m_i is the ith component of $\mathbf{m}_{\text{prior}}$. Note, 0.0004 was chosen so that there was some baseline for variance (when m_i is close to 0). Observe 0.15 was chosen due to the minimization of the misfit function (computed using the Gauss-Newton method) shown in Figure 5.3. For larger values of β (less regularization) we could achieve a better fit, but the change would be minimal. With more regularization, the a priori term begins to dominate.

Thus we have $\mathbf{m}_{\text{prior}}$ and $\mathbf{C}_{\text{M}}$ defined. Note $\rho_M(\mathbf{m})$ is 19-dimensional and hard to visualize. Following [7], we know the marginal distribution at each node is Gaussian with mean m_i and

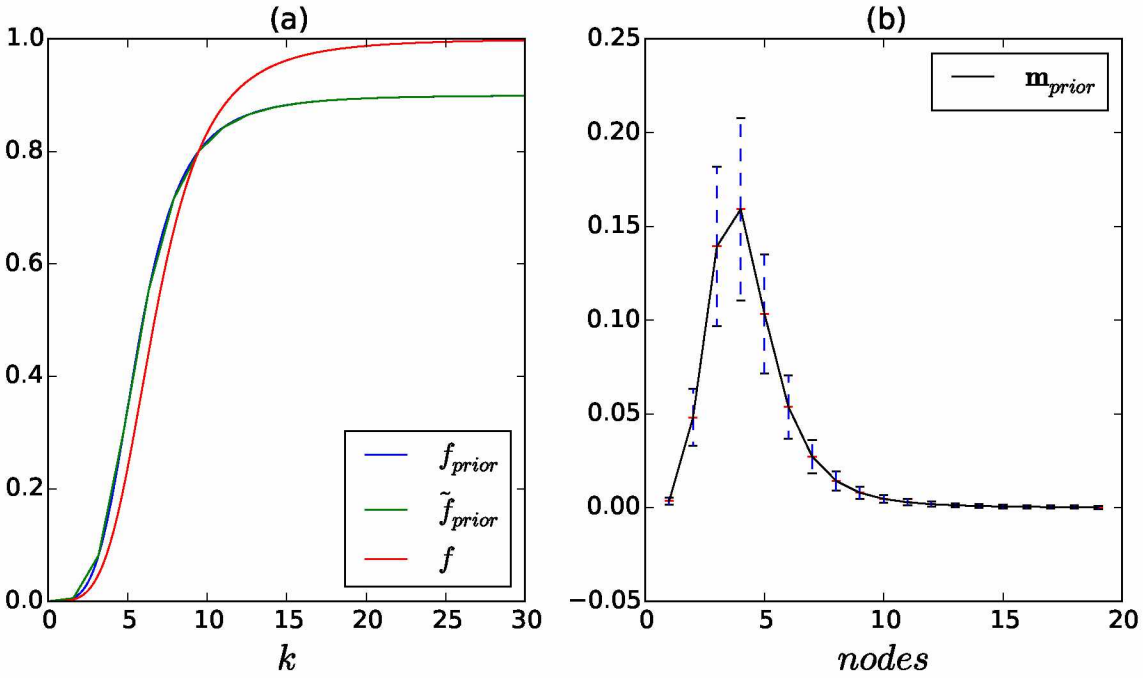

Figure 5.2: (a) Plot of f_{prior}, $\tilde{f}_{\text{prior}}$, and the production function which produced the unperturbed data. (b) Plot of $\mathbf{m}_{\text{prior}}$ with marginalized distribution at each node. The uncertainty bars represent a 95% credible interval.

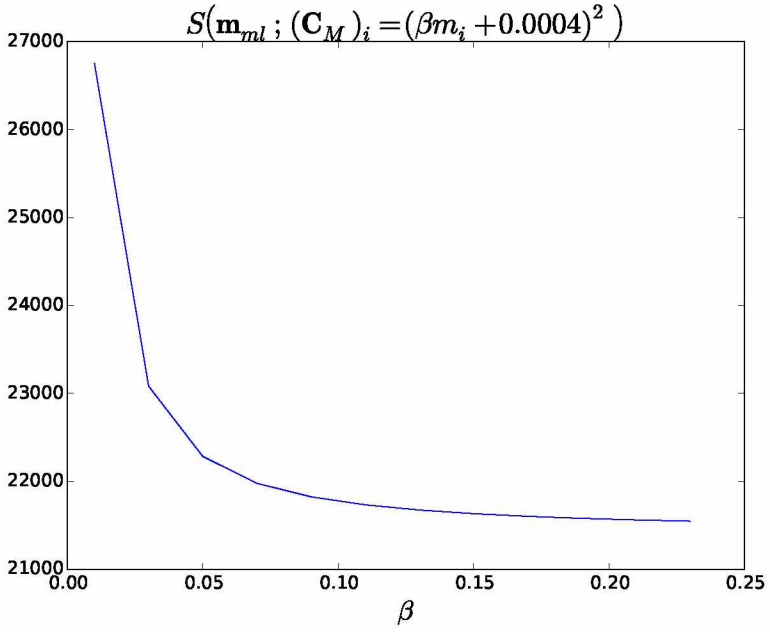

Figure 5.3: Evaluation of the misfit function depending on level of regularization.

standard deviation $(0.15m_i + 0.0004)$. We can then plot the components of $\mathbf{m}$ along with certainty bars which show a 95% credible interval (plus/minus 2 standard deviations). This visualization of the prior distribution is in Figure 5.2(b).

5.2 Specifications of the two approaches

Before we can apply the Metropolis algorithm to sample the posterior distribution and least squares linearize around $\mathbf{m}_{\text{MAP}}$, we need to specify the parameters for both the Metropolis algorithm and Newton's method.

The Metropolis algorithm defines an acceptance criteria to move to a new a point in a random walk. There are four necessary parameters needed to sample $\sigma_M(\mathbf{m})$ via Metropolis: the initial starting point $\mathbf{m}^{(0)}$, a way to take a proposed step (or stay put), the frequency which points are added to the sample points (to preserve memory), and the sample size.

The most logical place to start our random walk would be $\mathbf{m}_{\text{prior}}$, so let $\mathbf{m}^{(0)} = \mathbf{m}_{\text{prior}}$. From here we need to take a preliminary step. In (3.1) we proposed to perturb each node by a distance taken from normal distribution with standard deviation σ. A suggested way to determine step size given in [9] is by the acceptance rate of the Metropolis rule (3.2) - (3.3). As a rule of thumb, this acceptance rate should be roughly 30-50%. After experimenting (with various priors) for $n = 19$, we have determined to use $\sigma = 5 \times 10^{-5}$. For the computation here, this has an approximate acceptance rate of 39%.

Again though, we do not include all of these random walk points in the sample. We suspect a large number of samples will be needed; therefore, we will thin the samples by only including every fifth sample from the Metropolis algorithm.

We lastly need to determine at which iteration to start sampling and how large our sample size should be. Again, computational resources dictate this cannot be too large. We run the Metropolis algorithm until 125,000 samples are found (either by adding a new location or adding the current location). Of these, we will include every fifth sample. From looking at the trace plots of the parameter values at each iteration in Figure 5.4, it appears we should avoid using the first few thousand samples. We will then summarize our statistics after a burn in of 10,000 samples (note the dashed line in Figure 5.4). Therefore, the sample size from which we will determine the sample mean and credible intervals is 15,000. A larger

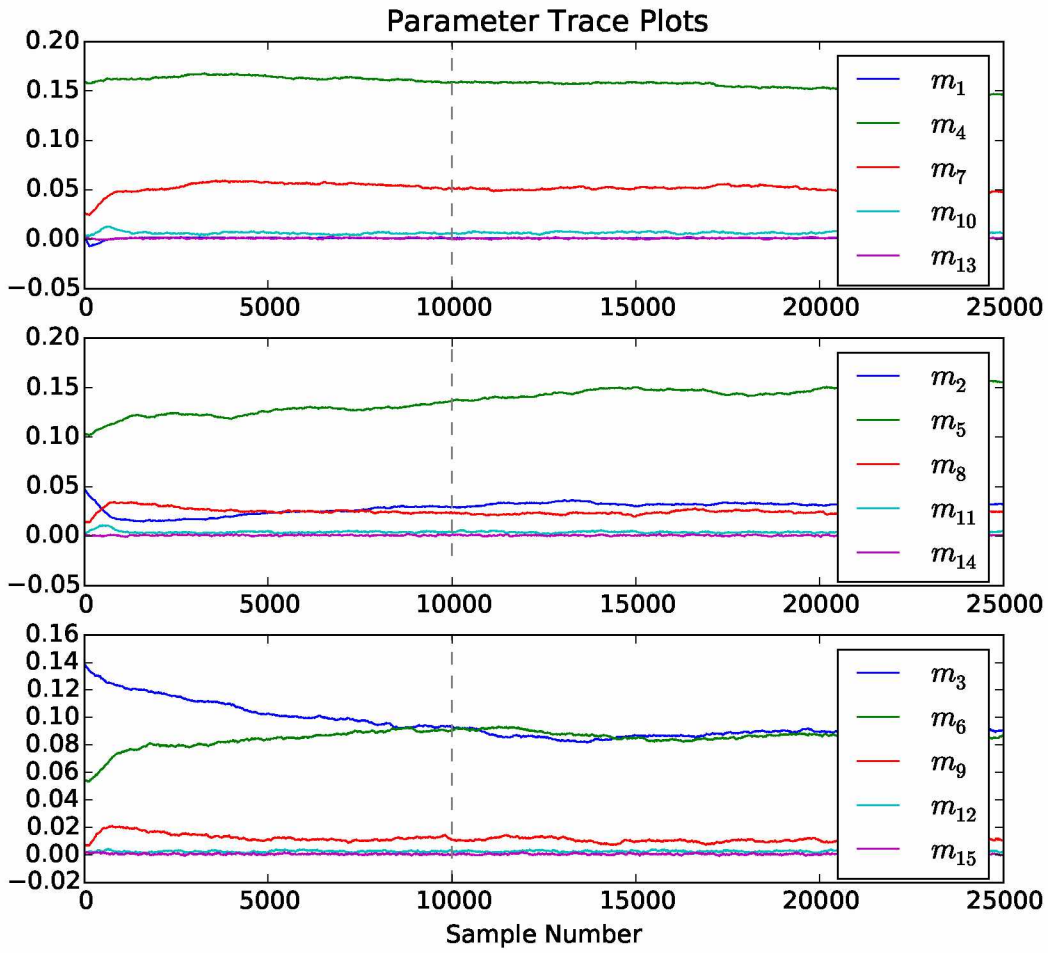

Figure 5.4: These three plots show the values for m_1 through m_{15}, the first 15 components of $\mathbf{m}$.

sample size could give a better description of our posterior distribution, but computing resources is the limitation here.

The parameters for the Gauss-Newton method described in Section 4.2 are easier to define. There are three of these, but as a contrast to the Metropolis approach two of them can be confidently chosen. The parameters are the initial starting point, the criteria for stopping, and, harder to define, the value for α in (4.4). We will again define $\mathbf{m}^{(0)} = \mathbf{m}_{\text{prior}}$. After experimenting with values for α, we chose $\alpha = 0.3$. A more robust solution for finding an iterate could come from a line search along the descent direction $\mathbf{s}$, but we have found $\alpha = 0.3$ to be small enough for convergence, but large enough for the efficiency of the algorithm.

We finally need to stop Newton's method. Finding $\mathbf{m}$ such that $\nabla S(\mathbf{m}) = 0$ is unreasonable. We should however stop when the gradient is "close enough" to 0. We will define close enough in a relative sense. We will stop Newton's method after j iterations where j is the smallest integer satisfying

$$\frac{\|\nabla S(\mathbf{m}^{(j)})\|_\infty}{\|\nabla S(\mathbf{m}^{(0)})\|_\infty} < 10^{-8}. \tag{5.7}$$

Now that all quantities and parameters have been defined, we will turn our attention to implementing Metropolis and Gauss-Newton to finally describe the posterior distribution.

5.3 Implementation

The codes solving the forward problem, the Metropolis algorithm, and Gauss-Newton method (and supporting functions) are all written using Python. The scripts are mostly written from scratch, apart from, for instance, functions for solving linear systems and generating random numbers from probability distributions.

With that in mind, computing the solution via Metropolis and Gauss-Newton is costly. For solving the forward problem $\mathbf{g}$, each time step requires, at its core, the solution of a system of 251 equations. But this forward problem solution is required at each proposed step that is fed into the Metropolis algorithm and for each component in computing the approximate gradient and Hessian at a point. Adding in the usual function evaluation and arithmetic, the codes for both methods are slow.

The code for Gauss-Newton is also limited by the fact that it doesn't scale. With a doubling of components of $\mathbf{m}$, the entries of the Hessian quadruple. Before we reach 19 nodes, this scaling problem becomes evident.

5.4 Comparison of solutions

We finally run both the Metropolis algorithm and Gauss-Newton following the specifications in section 5.2. As a result we obtain a sample $\{\mathbf{m}^{(i)}\}_{i=1}^{15000}$ of the posterior distribution (from Metropolis) and the maximum a posteriori estimate $\mathbf{m}_{\text{MAP}}$ and covariance $\tilde{\mathbf{C}}_{\text{M}}$ (from Gauss-Newton). Since $\mathbf{g}(\mathbf{m})$ is not linear, and in particular σ_M is not Gaussian, we don't expect the mean of this sample to coincide with $\mathbf{m}_{\text{MAP}}$. Nonetheless, with the samples $\{\mathbf{m}^{(i)}\}$, we find the sample mean $\hat{\mathbf{m}}$ by the following formula for $K = 15000$:

$$\hat{\mathbf{m}} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{m}^{(i)} \tag{5.8}$$

If σ_M is far from Gaussian, the sample covariance matrix computed from these samples may have little meaning [9]. Instead from these samples, we can find the middle 95% for each component of $\mathbf{m}$, and this will define the credible interval describing our certainty of the sample mean parameter. We then visualize this distribution marginalized at each node in Figure 5.5(b)

Since we also have $\mathbf{m}_{\text{MAP}}$ and $\tilde{\mathbf{C}}_{\text{M}}$ from least squares optimization, we at least have something to compare. We visualize the Gaussian distribution that approximates $\rho_M(\mathbf{m})$ in a neighborhood of $\mathbf{m}_{\text{MAP}}$ by again marginalizing at each node. We compute the standard deviation for each component by taking the square roots of the diagonals of $\tilde{\mathbf{C}}_{\text{M}}$. Figure 5.5(a) shows the marginal distributions of the approximate Gaussian at $\mathbf{m}_{\text{MAP}}$ where the certainty bars show plus/minus 2 standard deviations.

Figure 5.5 shows a discrepancy between the two solution methods: $\mathbf{m}_{\text{MAP}}$ does not coincide with $\hat{\mathbf{m}}$, and the uncertainty shown from least squares is greater than that shown from the samples of the a posteriori distribution. In either case, we can determine the piecewise production function and a description of how certain we are of said function. The piecewise production functions resulting from $\mathbf{m}_{\text{MAP}}$ and $\hat{\mathbf{m}}$ are shown in Figure 5.6(a) and Figure 5.6(b) respectively. The dashed curves in the figures show production functions prescribed by the credible intervals found from both methods.

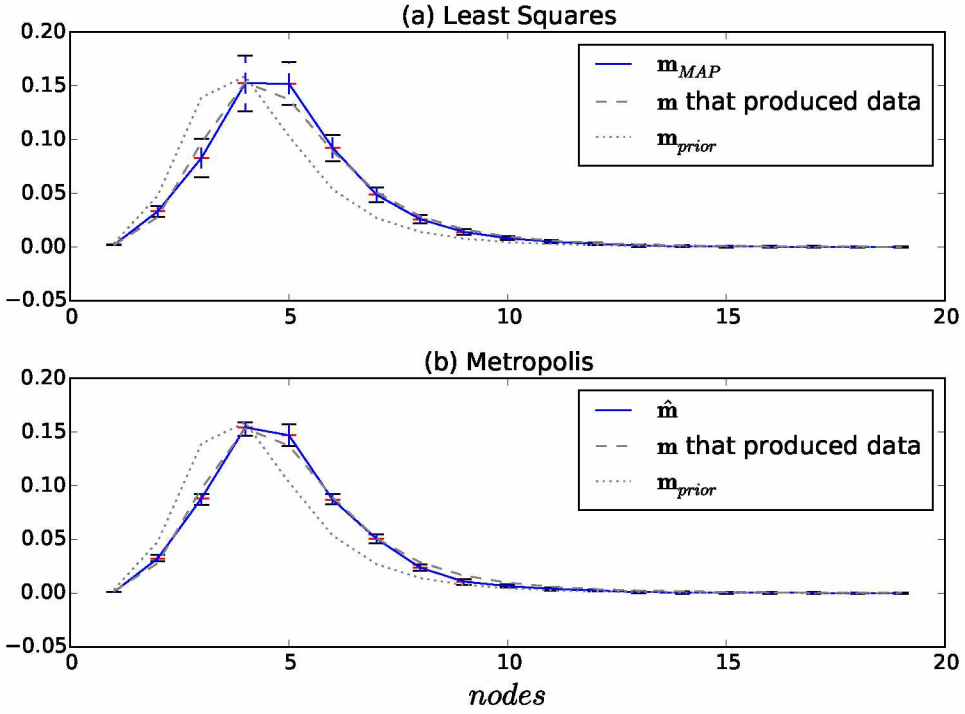

Figure 5.5: (a) Plot of $\mathbf{m}_{\text{MAP}}$ with credible intervals at each node. (b) Plot of $\hat{\mathbf{m}}$ with marginalized distribution at each node.

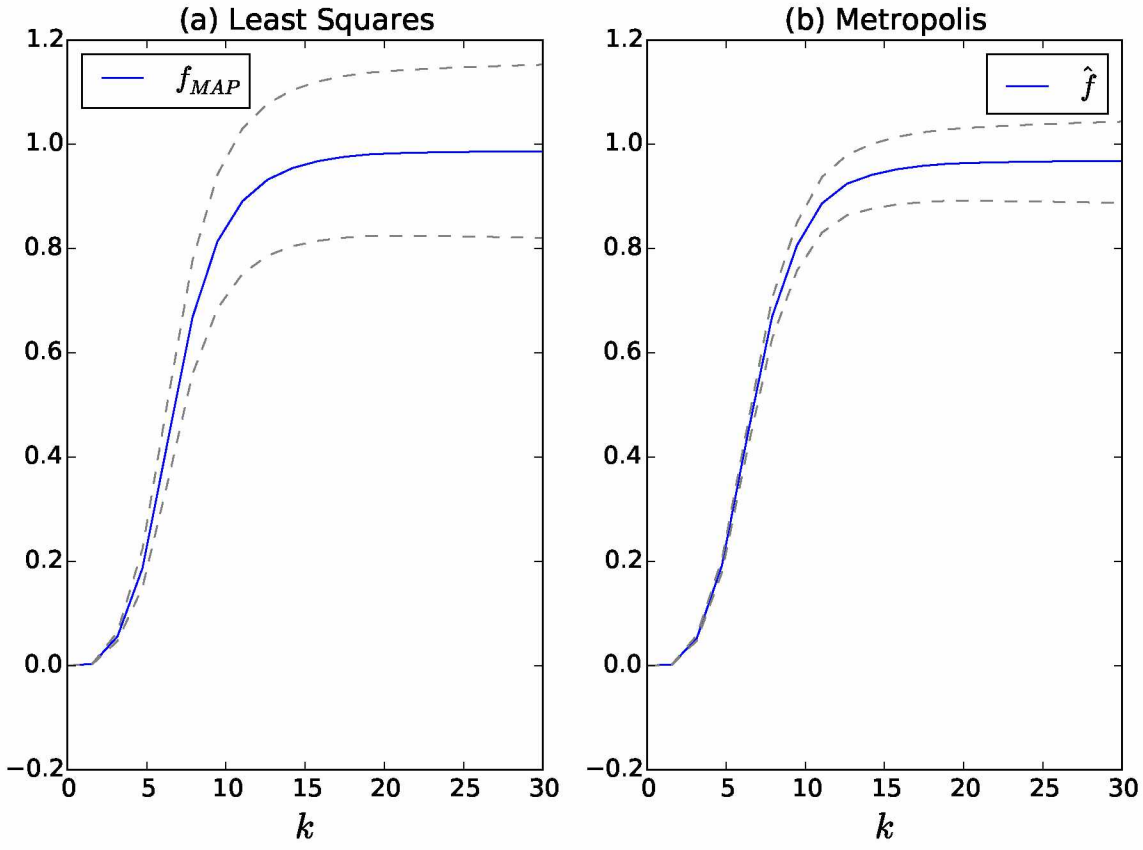

Figure 5.6: (a) Plot of production function from least squares; f_{MAP} determined by slopes $\mathbf{m}_{\text{MAP}}$ (b) Plot of production function from Metropolis algorithm; $\hat{f}$ determined by slopes $\hat{\mathbf{m}}$.

5.5 Remarks

There are a few potential reasons for the inconsistent answers. Beginning with the Metropolis algorithm, better results may be obtained by increasing the sample size. Since the samples are correlated (even with the thinning), a larger number is needed to better sample the posterior distribution. The only problem with this potential solution is the computational expense of running the Metropolis algorithm (and computing the forward problem). Another potential issue may come from approximating the gradient in the least squares method. That is, the approximation that results from approximating the derivative (Jacobian) of the forward model $\frac{\partial \mathbf{g}}{\partial \mathbf{m}}$.

Of course the summary statistics we are finding from each method come from *different* distributions. Metropolis samples the posterior distribution, but the covariance matrix $\tilde{\mathbf{C}}_{\mathrm{M}}$ comes from a Gaussian distribution that comes from linearizing the forward problem. There is a question of how close this distribution approximates the posterior distribution σ_M. This could account for the large difference in uncertainty bars.

In order to use actual data in solving this inverse problem (finding a production function f), we would need to solve the forward problem for two spatial dimensions (instead of just one). But then, a better model describing the diffusion of capital may be necessary.

References

[1] Ascher, U.M., Ruuth, S.J., Wetton, T.R.: Implicit-explicit methods for time-dependent partial differential equations. SIAM J. Numer. Anal. **32**.3, 797-823 (1995)

[2] Capasso, V., Engbers, R., La Torre, D.: On a spatial Solow model with technological diffusion and nonconcave production function. Nonlinear Analysis: Real World Applications **11**, 3858-3876 (2010)

[3] Engbers, R., Burger, M., Capasso, V.: Inverse problems in geographical economics: parameter identification in the spatial Solow model. Phil. Trans. of the Royal Society A **372**, 20130402 (2014)

[4] Gomez-Ramirez.: Don't blame the economists. It is an inverse problem! Eur. Journal of Futures Res. **1**:13, doi:10.1007/s40309-013-0013-6 (2013)

[5] Krugman, P., Wells, R.: *Macroeconomics 3rd Edition.* New York: Worth Publishers, 2012

[6] LeVeque, R.: *Finite Difference Methods for Ordinary and Partial Differential Equations.* Philadelphia: SIAM Press, 2007

[7] Mukhopadhyay, N.: *Probability and Statistical Inference.* New York: Marcel Dekker, 2000

[8] Nocedal, J., Wright, S.J.: *Numerical Optimization.* New York: Springer, 2006

[9] Tarantola, A.: *Inverse Problem Theory and Methods for Model Parameter Estimation.* Philadelphia: SIAM Press, 2005